(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 July 2001 (19.07.2001)

PCT

(10) International Publication Number
WO 01/52078 A1

(51) International Patent Classification⁷: G06F 15/00

(21) International Application Number: PCT/US01/01214

(22) International Filing Date: 12 January 2001 (12.01.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/483,439 14 January 2000 (14.01.2000) US

(71) Applicant: SCREAMINGMEDIA INC. [US/US]; 601 West 26th Street, New York, NY 10001 (US).

(72) Inventor: MCGINTY, Brian; Screamingmedia Inc., 601 West 26th Street, New York, NY 10001 (US).

(74) Agent: HANCHUK, Walter, G.; Morgan & Finnegan, L.L.P., 345 Park Avenue, New York, NY 10154 (US).

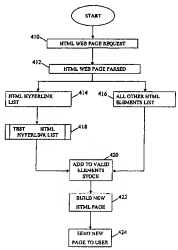
(81) Designated States (national): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW.

(84) Designated States (regional): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).

Published:
- with international search report

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: DEAD HYPER LINK DETECTION METHOD AND SYSTEM



(57) Abstract: A method and system for automatically checking the validity of hyperlinks embedded in web pages being served to clients by these servers. In response to a web page request, a server will parse the document (step 412) and separate the hyperlinks from the other elements in the documents (steps 414, 416). The server will then review the hyperlinks to determine whether "dead links" are present (step 418). The server will then either remove the dead link or will strip away the tags that empower the link thus making the link look like plain text. The server will then reconstruct the document including the hypertext links and other elements and send the document to the requestor (steps 422, 424).

WO 01/52078 A1

DEAD HYPER LINK DETECTION METHOD AND SYSTEM

Field of Invention

The present invention relates generally to the field of document retrieval and interaction on a distributed computer network. More specifically, the present invention relates to a system for post processing embedded hyperlinks.

Background of the Invention

The World Wide Web (WWW) may be broadly described as a virtual collection of documents with a user being able to access and retrieve these documents through existing telephone or data lines. Documents accessible on the WWW have the capability to direct users to other documents on the web using linking information imbedded in the text itself. Typically, the documents are stored in hypertext markup language (HTML) format. Using hypertext linking an author will integrate references directly into the text of a document which point to other related items of information. Uniform resource locators (URLs) provide a way of converting the integrated reference to a real location where the related information will be located on the Internet. It is possible that links that are valid when they are included in these pages may become defunct or "dead links" over time.

Summary of the Invention

An aspect of the present invention involves a method of testing embedded hyperlinks including receiving a document request from a client; parsing a first document to determine if elements in the first document contain hyperlinks; separating the elements into hyperlinks and all other non-hyperlink elements; testing the hyperlinks in a first document in parallel to determine if the hyperlinks are valid hyperlinks or invalid hyperlinks by comparing the hyperlinks to a predetermined rule set; adding the valid hyperlinks to a list including the other non-hyperlink elements; generating a second document from the list; and providing the second document to the client.

Another aspect of the present invention involves a system including a memory device which stores a first document; and a processor in communication

with the memory device, said processor configured to: receive a document request from a client; parse the first document to determine if elements in the first document contain said hyperlinks; separate the elements into hyperlinks and all other non-hyperlink elements; test hyperlinks in said first document to determine if said hyperlinks are valid hyperlinks or invalid hyperlinks by the comparing the hyperlinks to a predetermined rule set; add the valid hyperlinks to a list including the other non-hyperlink elements; generate a second document using the list; and provide said second document to said client.

Other and further aspects of the present invention will become apparent during the course of the following description and by reference to the attached drawings.

Brief Description of the Drawings

Figure 1 illustrates a block diagram of an internet client/server relationship;

Figure 2 illustrates a block diagram of the server of Figure 1;

Figure 3 illustrates an HTML document in an exploded view;

Figure 4 illustrates a flow chart of the process of link validation of an embodiment of the present invention;

Figure 5 illustrates a first subroutine of the flow chart of Figure 4 in which the hypertext links and other text are separated;

Figure 6 illustrates a second subroutine of the flow chart of Figure 4 in which the hypertext links are tested to determine if they are valid; and

Figure 7 illustrates an alternative embodiment of the present invention which includes a modification of the subroutine of Figure 5 so that invalid hypertext links are processed to strip away the HTML tags .

Detailed Description of the Preferred Embodiments

The ability of a web server application to ascertain the validity of embedded links in web pages at request time is critical for the creditability of a web site. With more and more web sites moving into the e-commerce arena, this question of web site creditability is becoming even more sensitive. The present invention is capable of detecting defunct hyper links as soon as they become accessible.

Embodiments of the present invention disclosed herein relate to the serving of web pages or documents by Internet web servers. The pages or documents discussed in this application may be in Hyper Text Markup Language (HTML), Standard Generalized Markup Language (SGML), Extensible Markup Language (XML) or any other format which uses a tagging architecture. In the following discussion of this application, HTML will be used for example purposes only.

The embodiments disclosed herein include a method and system for checking the validity of HTML hyperlinks embedded in HTML web pages being served to clients by a server. This is true of web servers that serve static (or non-changing HTML web pages) or application web servers that serve dynamic HTML web pages. Static web pages are HTML web pages that are written or "constructed" at some point in time and then remain unchanged until a web site administrator manually either removes them, updates them, or replaces them with entirely new pages. Dynamic HTML web pages are web pages served through some type of application server utilizing HTML templates and some type of dynamic page generation mechanism. In both cases it is possible that links that are valid when they are included in these pages may become defunct or "dead links" over time.

With reference to the Figures, several embodiments of the present invention will now be shown and described. Referring to Figure 1, electronic content distribution system 100 includes a server 110 and a user computer/client 140 both of which are connected across network backbone 105. Network backbone 105 may include an internet backbone, an intranet backbone or any other conventional network backbone or a combination thereof.

Server 110 may be a conventional server which includes conventional computer hardware and functionality. Server 110 may be associated with a web site or a content provider, such as a publisher (e.g., a magazine publisher, book publisher, etc.), a news agency, or any distributor or provider of electronic content. Electronic content may correspond to any publications (e.g., a news or magazine article), reports, technical papers and so forth. Electronic content may include a content body including documents with text and/or images with associated meta-data as well as traditional index fields generally provided in a header or trailer

section of this electronic content. Server 110 is configured to perform automatic dead link checking of hyperlinks to determine if dead links appear in a content body of the electronic content.

Fig. 2 is a schematic block diagram illustrating the components of server 110 of Fig. 1. Conventional computer components are included, such as a processor 200, user input devices 205, e.g., keyboard, mouse, etc., for receiving user inputs, network interface 210 for interconnection to the network backbone 105, RAM 215, ROM 220, display 225 and storage device 230. Storage device 230 stores the software which implements the present invention.

Turning to Figure 1, a request is sent from user computer 140 onto the network backbone 105 for a particular document or other piece of information. The requested document 320 as shown in Figure 3 is stored on server 110. The document 320 may include highlighted text 322 which includes hidden embedded links to other related information as prepared by hypertext authoring tools. The present invention will automatically perform a dead link check on any hyperlinks in the document 320 before sending the document to the user computer 140.

Figure 4 illustrates a flow diagram of the elemental steps of a first embodiment of the present invention. In a first step 410, a user accesses an Internet resource, such as an HTML page, which is served by the server 110. In step 412, the server 110 will, before serving the page to the user, parse that page and isolate the HTML hyper links that are embedded in that page. Figure 5 illustrates step 412 in more detail. In step 412a, a comparison is performed between the HTML page and a predefined rule set. Since all HTML hyperlinks employ a defined syntax the server 110 can work from this predetermined rule set for parsing and isolating these links. This predetermined rule set can optionally be augmented through the use of a web server configuration file. This configuration file may employ an HTML hyper link meta language that will allow the server 110 to dynamically learn at initialization time the syntax and nature of the HTML hyperlinks that must be isolated. In step 412b, a decision is made whether the text is a hyperlink. If so, it is added to the list of "N" hyperlinks in 412c (with N representing a number greater than or equal to 0). If the text is not a hyperlink, it is added to the list of all other

HTML elements which are not hyperlinks 412d. In step 412e, the system determines if all of the document has been checked and if not, returns to step 412a to continue checking the document. If the entire document has been reviewed, then the hyperlink parsing is completed in step 412f and the program returns to the flowchart of Figure 3.

Figure 4 shows that in steps 414 and 416 the hyperlink list of the "N" links and the other non-hyperlink HTML elements lists are separated. Once the server 110 has isolated the list of hyperlinks for a given web page it may in step 418 employ a multi-threaded socket initiator to simultaneously create hypertext transfer protocol (HTTP) socket connections to all the hyperlinks in the hyperlink list and allow the hyperlinks to be tested in parallel. These socket connections will begin retrieving the specified web pages looking in particular for web server error messages in HTTP headers of the incoming pages. For example a 404 return code signifies that the web page in question no longer exists at the specified location. Once the HTTP header is read, the socket connection may be terminated. It is then a matter of parsing and interpreting the headers for the various web pages.

Figure 6 discloses step 418 in more detail. In step 418a, hyperlinks 1 to N are tested. If the first through "N" hyperlinks are valid as determined in steps 418a through 418c then these hyperlinks are given a Boolean value of VALID and added to the list of valid hypertext links in 418d. If these hyperlinks are not valid, then the hyperlink is given the Boolean value of NOT VALID and not added to the list of valid hyperlinks and the program returns to the flowchart of Figure 4.

At this point the server 110 has the HTML web page parsed into a dynamic data structure with the hyperlinks separated from the remaining page elements. The server 110 also has a dynamic data structure that has a list of the pages internal links and a Boolean value that represents that links web status (i.e., VALID or NOT VALID). The server 110 will recombine the VALID hyperlinks with the other HTML elements in step 420 and omit any hyperlinks having a NOT VALID value. The server 110 will recompose the elements of the page in step 422. In this way the user will never see invalid or defunct links being served by the web site that employs a server 110 such as this.

In an alternative embodiment disclosed in Figure 7, subroutine 418 will be modified so that server 110 will recompose the page with the non-valid link but will strip away the HTML tags that empower that link, thus making the link look like plain text. In this embodiment, the net result is the same. A user will never click on a hyper link that takes them to a defunct page. Subroutine 418 will be modified to include steps 418e through 418g in which if a hyperlink is found to be invalid, the tag will be stripped and the link will be made to look like text and added to VALID hyperlink list.

The foregoing is to be understood as being in every respect illustrative and exemplary, but not restrictive, and the scope of the invention disclosed herein is not to be determined from the Detailed Description, but rather from the claims as interpreted according to the full breadth permitted by the law. It is to be understood that the embodiments shown and described herein are only illustrative of the principles of the present invention and that various modifications may be implemented by those skilled in the art without departing from the scope and spirit of the invention.

Claims

1. A method of testing embedded hyperlinks comprising:

receiving a document request from a client;

parsing a first document to determine if elements in the first document contain hyperlinks;

separating the elements into hyperlinks and all other non-hyperlink elements;

testing the hyperlinks in a first document in parallel to determine if said hyperlinks are valid hyperlinks or invalid hyperlinks by comparing the hyperlinks to a predetermined rule set;

adding the valid hyperlinks to a list including the other non-hyperlink elements;

generating a second document from said list; and

providing said second document to said client.

2. A method comprising:

automatically testing hyperlinks in a first document to determine if said hyperlinks are valid hyperlinks or invalid hyperlinks; and

generating a second document using the valid hyperlinks.

3. The method of claim 2, further comprising:

stripping tags from the invalid hyperlinks and adding the invalid hyperlinks to the second document.

4. The method of claim 2, wherein said testing of the hyperlinks is performed in parallel.

5. The method of claim 2, further comprising:

receiving a document request from a client; and

providing the second document to the client.

6. The method of claim 2 further comprising:

parsing the first document to determine if elements in the first document contain said hyperlinks.

7. The method of claim 2 further comprising:

separating the hyperlinks from other elements in the first document; and

adding the valid hyperlinks to the other elements before generating said second document.

8. The method of claim 2, wherein said parsing step includes comparing said elements to a predetermined rule set.

9. The method of claim 2, wherein said first and second documents are static web pages.

10. The method of claim 2, wherein said first and second documents are dynamic web pages.

11. The method of claim 2, wherein said first and second documents are written in a format from one of the group consisting of HTML, SGML, and XML.

12. The method of claim 2, further comprising:

stripping tags from the invalid hyperlinks and adding the invalid hyperlinks to the list.

13. A system comprising:

a memory device which stores a first document; and

a processor in communication with said memory device, said processor configured to:

automatically test hyperlinks in said first document to determine if said

hyperlinks are valid hyperlinks or invalid hyperlinks; and

generate a second document using the valid hyperlinks.

14. The system of claim 13, said processor further configured to:

strip tags from the invalid hyperlinks and add the invalid hyperlinks to the second document.

15. The system of claim 13, said processor further configured to:

test said hyperlinks in parallel.

16. The system of claim 13, said processor further configured to:

parse the first document to determine if elements in the first document contain said hyperlinks.

17. A system comprising:

a memory device which stores a first document; and

a processor in communication with said memory device, said processor configured to:

receive a document request from a client;

parse the first document to determine if elements in the first document contain said hyperlinks;

separate the elements into hyperlinks and all other non-hyperlink elements;

test hyperlinks in said first document to determine if said hyperlinks are valid hyperlinks or invalid hyperlinks by the comparing the hyperlinks to a predetermined rule set;

add the valid hyperlinks to a list including the other non-hyperlink elements;

generate a second document using the list; and

provide said second document to said client.

18. A system comprising:

means for automatically testing hyperlinks in a first document to determine if said hyperlinks are valid hyperlinks or invalid hyperlinks; and

means for generating a second document using the valid hyperlinks.

19. The system of claim 18, further comprising:

means for stripping tags from the invalid hyperlinks and adding the invalid hyperlinks to the second document.

20. The system of claim 18, further comprising:

a means for parsing the first document to determine if elements in the first document contain hyperlinks.

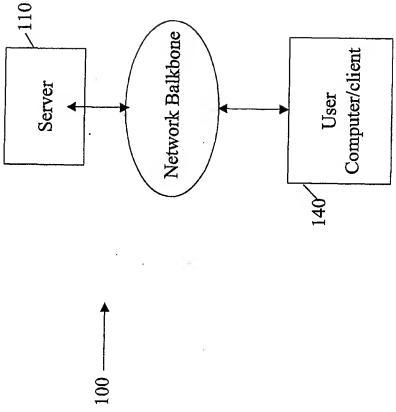


FIG. 1

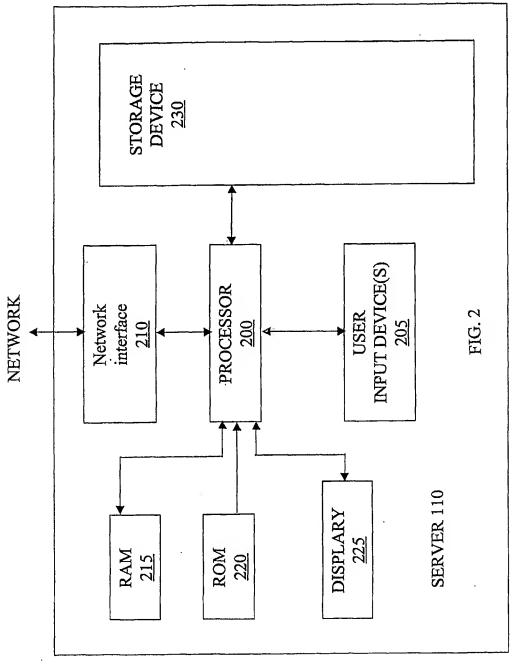

NETWORK
Network interface 210
PROCESSOR 200
STORAGE DEVICE 230
USER INPUT DEVICE(S) 205
RAM 215
ROM 220
DISPLARY 225
SERVER 110

FIG. 2

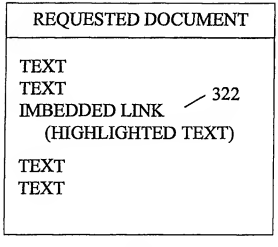

REQUESTED DOCUMENT
TEXT
TEXT
IMBEDDED LINK
322
(HIGHLIGHTED TEXT)
TEXT
TEXT
320

FIG. 3

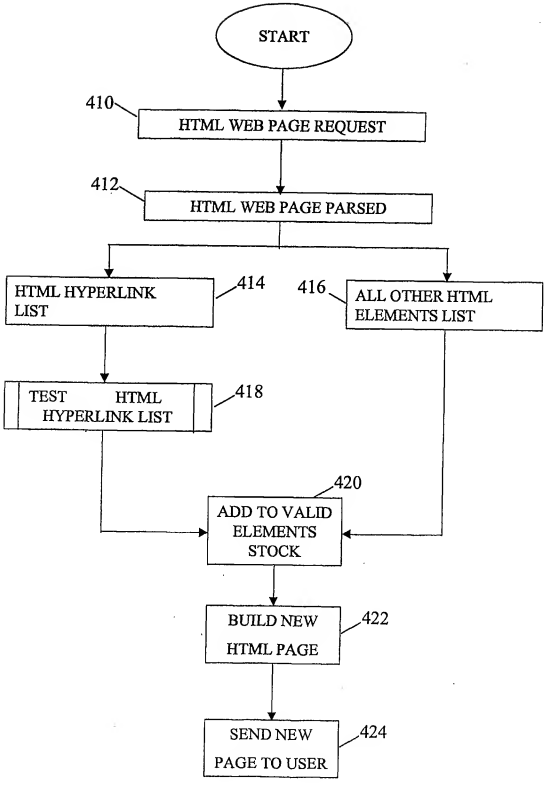

START
410
HTML WEB PAGE REQUEST
412
HTML WEB PAGE PARSED
HTML HYPERLINK LIST
414
416
ALL OTHER HTML ELEMENTS LIST
TEST HTML HYPERLINK LIST
418
420
ADD TO VALID ELEMENTS STOCK
BUILD NEW HTML PAGE
422
SEND NEW PAGE TO USER
424

FIG. 4

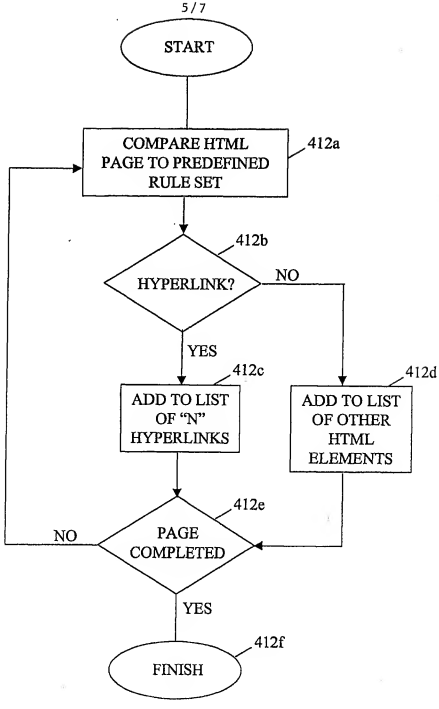

START
COMPARE HTML PAGE TO PREDEFINED RULE SET
412a
412b
HYPERLINK?
NO
YES
412c
ADD TO LIST OF "N" HYPERLINKS
412d
ADD TO LIST OF OTHER HTML ELEMENTS
412e
PAGE COMPLETED
NO
YES
412f
FINISH
412

FIG. 5

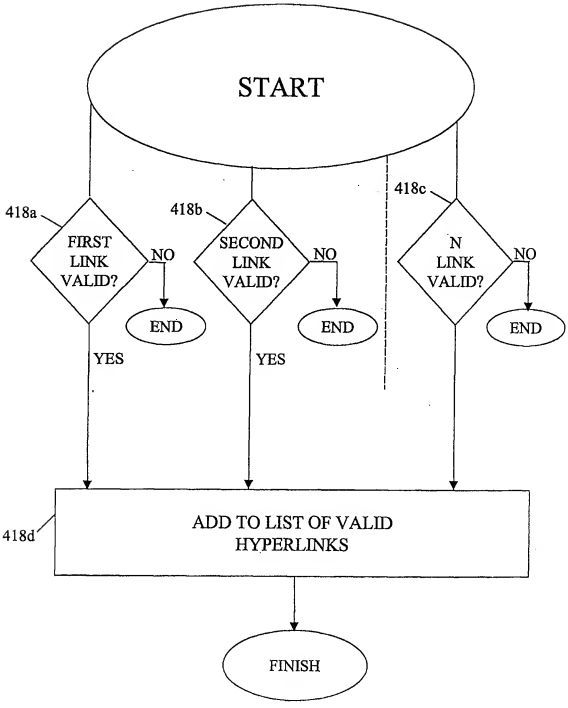

START
418a
FIRST LINK VALID?
NO
END
YES
418b
SECOND LINK VALID?
NO
END
YES
418c
N LINK VALID?
NO
END
418d
ADD TO LIST OF VALID HYPERLINKS
FINISH

FIG. 6

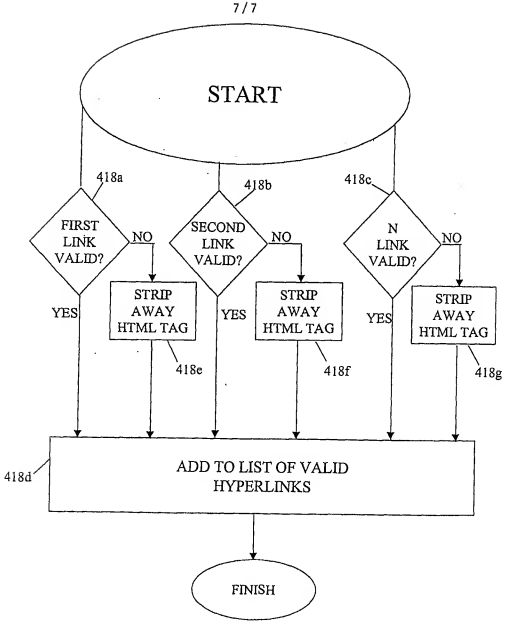

START
418a
FIRST LINK VALID?
NO
YES
STRIP AWAY HTML TAG
418e
418b
SECOND LINK VALID?
NO
YES
STRIP AWAY HTML TAG
418f
418c
N LINK VALID?
NO
YES
STRIP AWAY HTML TAG
418g
418d
ADD TO LIST OF VALID HYPERLINKS
FINISH

FIG. 7

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US01/01214

A. CLASSIFICATION OF SUBJECT MATTER

IPC(7) :G06F 15/00
US CL :707/513

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 707/513, 514

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

IEEE

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y, P	US 6,035,330 A (ASTIZ et al.) 07 March 2000, figure 10, col 12, lines 1-55.	1-2,4-11,13 15-18, 20.
Y	US 5,995,099 A (HORSTMANN) 30 November 1999, col 6, lines 23-67.	3, 12, 14, 19
Y	LEUNG, A Tool for Testing Hypermedia Systems, abstract, page 203.	1-2, 4-11, 13-18, 20
A	SCOTTS, Petri-Net-Based Hypertext: Document Structure with Browsing Semantics, introduction, pages 9, 21.	1-20

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance		
"E"	earlier document published on or after the international filing date	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed	"&"	document member of the same patent family

Date of the actual completion of the international search 09 MARCH 2001	Date of mailing of the international search report 09 APR 2001
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer HEATHER HERNDON Telephone No. (703) 308-5186

Form PCT/ISA/210 (second sheet) (July 1998)*